Inferring model structures from inertial sensor data in distributed activity recognition scenarios

Pierluigi Casale and Oliver Amft

ACTLab, Signal Processing Systems, TU Eindhoven
Eindhoven, The Netherlands
{p.casale,o.amft}@tue.nl
http://www.actlab.ele.tue.nl

Abstract. Activity-Events-Detectors digraphs describe the relations between human activities and sensor nodes under a distributed perspective. The graphs provide a conceptual abstraction that decouples the set of activities from the sensor network with the aim of improving the recognition performances and lowering the computational constraints of the detection tasks in the sensor node. In this work, a data-driven methodology that learns groups of activities and infers the configuration of detectors embedded in the sensors nodes of the network is proposed. The methodology, defined on a clustering procedure, derives and infers all the relevant information from sensors data, making no a-priori assumptions on the relations between sensor nodes and activities. Using the inferred structured models, a performance boost of 15% in the final classification accuracy is obtained with a significant reduction of the computational resources needed for recognition purposes.

Key words: context recognition, wireless sensor networks, clustering, inertial sensors

1 Introduction

The recognition of human activities from raw sensor data finds one of its mayor application in Ambient Intelligence scenarios. Automatic activity monitoring enables the development of personalized ubiquitous services in health-care and assisted living domains [1] providing, at the same time, more natural interactions in smart homes and smart environments [2]. Although different types of sensor modalities are effective for classifying different activities, a sensors network gathering data from different objects and locations is potentially able to identify a large set of heterogeneous activities. This setting naturally envisions a recognition architecture where the set of non-divisible pieces of information that each node can sense constitutes the alphabet of elements of complex activities.

The Activity-Events-Detectors (AED) [3] paradigm describes these concepts in a formal way. Based on a graph-based formalism, the paradigm shows the relationships between activities and sensor nodes under a distributed perspective, following the model of hierarchical activity recognition. The formalism describes the dependencies between the activities that can be observed by each node, the detection performed at sensor level and the physical nodes, providing a conceptual abstraction where the full set of activities is decoupled from the sensors network. The decoupling is provided by *events*, groups of atomic components that are recognized by sensor nodes from their local sensing modalities. The task of a detector is the extraction of valid atomic activities from the continuous data stream and the disambiguation of the events discriminable by the detector. This grouping reflects the local position and the sensing capabilities of the node. The paradigm provides significant advantages in terms of transmission bandwidth since, instead of sending raw or compressed data, only the detection outcome is transmitted over the network. Moreover, dynamic reconfiguration of the network can be allowed with significant benefits in the power consumption of the whole network.

Previous works ([4],[3]) show that, using domain knowledge, detectors can be manually configured by grouping atomic activities into events providing a valuable enhancement in the recognition performance of the architecture. In this paper, a data-driven methodology that infers the model structure of detectors under the AED framework is presented and validated. Given a set of atomic activities, the proposed approach automatically learns groups of patterns that look similar from the local perspective of the node and are relevant for the final recognition of composite activities. The methodology, defined on a non parametric clustering procedure that uses the classification margin as patterns similarity measure, makes no assumptions about the number of events and the relations between detectors and activities: all the relevant information are derived and inferred from the data. The approach has been compared with manual detector configurations provided by experts and with configurations generated using the K-Means clustering methodology. Results obtained show that the methodology is able to retrieve relevant activities for all the detectors and grouping them consistently to the manual configuration provided by experts. In particular, for the majority of the detectors, the proposed methodology is able to retrieve relevant activities for sensor nodes as annotated by experts. Moreover, groups of activities not annotated in the manual configuration are also generated by the methodology. These groups significantly help in the final recognition of composite activities. Recognition performance shows that the use of the AED paradigm significantly improves the classification performance in the layered recognition architecture. In particular, detectors performance are considerably increased and the final performance of the recognition architecture has a boost of 15% in terms of classification accuracy. Furthermore, the detector configuration provides a reduction of the computational resources needed by the detector nodes of more than 90% for all the sensor nodes of the network.

The rest of the paper is organized as follows. In Section 2 an overview of related works in hierarchical and distributed activity recognition is provided. In Section 3, a formal description of the Activity-Events-Detectors paradigm is reported and in Section 4, the proposed configuration methodology is presented and described in details. Section 5 presents the validation protocol alongside the description of the dataset used. Experimental results are presented in Section 6. Finally, Section 7 concludes the paper.

2 Related Works

Many works address the necessity of using hierarchical abstractions for modeling human activities. Aggarwal ee al. [5] provide a detailed overview of activity recognition research works with particular focus on hierarchical methodologies. Approaches are differentiated between statistical and syntactic approaches. Statistical approaches construct state-based models hierarchically concatenated, like layered hidden Markov models, to represent and recognize high-level human activities. Similarly, syntactic approaches, as in [6], use grammar syntaxes to model high-level activity as a string of atomic-level activities that sequentially compose complex human activities, allowing the generation of information fusion methodologies. Zappi et al. [7] investigate distributed information fusion using multiple classifiers strategies from sensors distributed on the body. Using classifiers fusion, recognition accuracy can be significantly boosted using clusters of sensors. This distributed scheme, that shares with the AED framework the idea of aggregating the detection results, does not consider the simplification of the detector by aggregating atomic activities at detector level. Sarkar et al. [8] consider the possibility to identify key sensors in the network that are related to different activities. Clusters of sensors are defined based on the activations that sensors have during similar activities. Key sensors are identified based on the number of activations per sensor. The proposed methodology provides significant improvement in terms of activity classification accuracy. Nevertheless, although the work share with the AED framework the idea of grouping similar activities, it still does not consider the grouping of those activities at sensors level. Storf et al. [9] describe an activity recognition architecture, where complex activities are recognized using a low-level activities decomposition. Atomic and complex activities are detected by a specialized detection agents that communicate by exchanging typed facts represented in a common data structure. The approach brings several practical advantages especially in terms of execution performance since only those parts of the overall functionality are invoked that are actually affected. Van Kasteren et al. [10] propose a two-layer hierarchical model with activities consisting of a sequence of actions where sensor data are automatically clustered during the training phase. Results obtained outperforms the non-hierarchical model providing the advantages of making easier the activity annotation process. The approach, that share with the AED paradigm the underlying idea of grouping similar activities at sensor level, does not provide any information about which are the activities grouped and, in particular, the clusters found do not have any meaningful correspondence to actual actions.

3 The Activity-Events-Detectors framework

Human activities are often considered under a hierarchical perspective in order to manage their inherent complexity. The correspondent recognition architecture can be usually modeled in a layered organization were a set of non-dividable unities, the *atomic activities* processed and identified from raw sensor data, is considered at the lowest level. At higher levels, the atomic activities are agglomerated into more complex sequences of activities. Accordingly, the Activity-Events-Detectors (AED) paradigm considers a set of distributed sensing and detection nodes that sense the contextual sensors data, identifies pattern events in the acquired sensor data and communicates the results of the detection for further processing. Often, the patterns of two or more atomic activities observed by detectors may look similar. In this case, several atomic activities can be mapped into events representing the final detection result of the nodes. Detected events are communicated among the distributed detector nodes such that they can be further processed. In a formal description, the set of composite activities, represented by the alphabet C, measures the number of composite activities that the system is able to recognize. The set of atomic activities A describes the detection alphabet and each composite activity C_n is composed by a subset of unique atomic activities from A.

$$C = \{C_1, ..., C_n\},\ A = \{a_1, ..., a_m\}\ C_k \subseteq A \tag{1}$$

Each detector node D_i contains at least one detector event E_i, as described in Eq. 2. The number of detector nodes $|D|$ and the total number of detector events $\sum_{i=1}^{|D|} |D_i|$ are complexity metrics for the implemented architecture.

$$D_i = \{E_1^i, ..., E_t^i\} \tag{2}$$

When atomic activities cannot be completely discriminated by a detector, the affected activities are grouped into one event of the detector: for each detector D_i, atomic activities a_j conflicting with each other are grouped to a single event E_j^i.

$$E_j^i \subseteq A,\ where\ \forall i : E_j^i \cap E_k^i = \emptyset,\ for\ j \neq k \tag{3}$$

The combination of multiple, distributed event detectors is used to recognize composite activities. The event-based composite activity C_n consists of a subset of events reported by different detectors D_i, where the set is empty, if the detector does not contribute to the recognition.

$$C^n = \bigcup_i D_n^i, \forall i : D_n^i \subseteq D^i \tag{4}$$

These relationships can be mapped in the form of directed bipartite graphs, as shown in Fig. 1. Two directed graphs are presents. The AE graph maps atomic activities into events, the ED graph maps events into atomic activities. Graph nodes of A are connected to graph nodes in E. No directed edges from

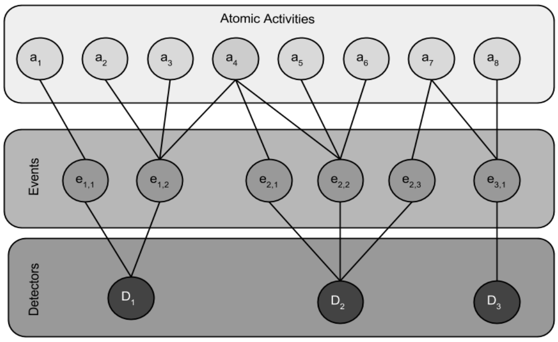

Fig. 1: Example AED digraph. Elements a_k correspond to atomic activities, e_j to events recognized, and d_i to detectors. Detectors represent sensors embedded in objects or infrastructure provided with computational power. High level activities are composed by groups of atomic activities.

E to A are present. Similarly, nodes in E are connected to nodes in D and no reverse edges are allowed. Graphs AE and ED may not have cycles or loops. The AED digraph helps in visualizing the main concepts of the AED paradigm showing which are the atomic activities that constitute events in each detector. This configuration, that in this work is aimed to be automatized, allows to establish which are the relevant atomic activities for each detector and, in particular, which are the groups of activities that the detector is able to discriminate, yielding to the advantages of using the AED formalism. In the following section, a data-driven methodology that automatically generated the detector configuration is presented. The methodology is decomposed in three steps: (i) identifying patterns of atomic activities that look similar from the local perspective of the detector, (ii) grouping them into events, and (iii) reject groups of activity patterns irrelevant for the detector.

4 Inference Methodology for Detectors Configuration

Grouping similar patterns is the task performed by clustering algorithms. The aggregation is computed on the base of a predefined similarity measure between data points. The classical K-means algorithm performs this procedure using a two steps iterative process which maximizes both the similarity between points belonging to the same cluster and the dissimilarity between clusters. The groups so found are shaped in a predefined number of convex clusters [11]. For many clustering methodologies, the number of clusters is a parameter for the algorithm. When no a-priori knowledge about the number of clusters is available, different grouping methodologies should be considered.

The main steps of the proposed configuration methodology are shown in Fig.2. In the first step, a proximity measure between activities is computed. Since activities are generally described by multiple features, this problem can be identified

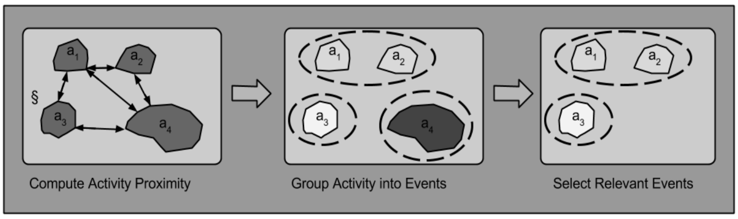

Fig. 2: Inference Methodology: Steps in the Detectors Configuration

as a geometric proximity problem. The resolution of this problem implies the identification of a boundary region of the data-points in the features space and the search of the closest points between all the neighbor activities. When low-dimensional spaces are considered, efficient solutions can be applied. However, the running time and memory requirements of these algorithms grow exponentially with the dimension of the features space. For the purposes of this work, an estimation of the proximity between activities is computed using the classification margin of all the pair-wise combinations of activities. The advantage of using this approach is twofolds: (i) the classification margin represents a noise-tolerant proximity measure since it does not depends by specific data-points and (ii) its computational burden is limited when compared to a high-dimensional geometrical proximity problems. Using the proximity measure provided by the classification margin, groups of activities are found using a clustering procedure based on Minimum Spanning Tree. This step generates a linkage between activities that minimizes the overall classification margin through the graph. The final grouping is obtained cutting the edges in the graph with highest margin. In this way, groups of activities that correspond to the *events* are discovered. However, not all the discovered events are relevant for the specific detector. As final step, the relevance of the events generated is quantified using a ranking procedure based on features selection, that removes events that are irrelevant for the detector. The overall procedure, resembling a single linkage agglomerative clustering algorithm, has a peculiar advantage. While the single linkage clustering works on actual data-points, the proposed methodology provides an high-level grouping of the activities that does not depend directly by activity data-points. In addition, the result of the whole procedure provides groups of activities that make no assumptions regarding the underlying activities distribution. In the following subsections, each step is described and explained in detail.

4.1 Compute Activity Proximity

Method: The classification margin is a measure of confidence in the classification process between two sets of patterns. In the simplest case of pairs of atomic activity patterns $\{a_i, a_j\}$ that are linearly separable, the classification margin is provided by the shortest distance of the closest examples from the separating hyperplane. This basic geometrical consideration can be extended for handling

the case when maximum margin hyperplanes are difficult to find due to noisy data [12], and the classification problem is not linearly separable [13]. Nevertheless, the quantity is still able to provide an approximate measure of proximity between patterns. Given a classification function f able to compute the margin, a general algorithm for computing the pairwise classification margins between atomic activity patterns is defined in Algorithm 1. For each pair of activity patterns, the classification function f is trained on the dataset constituted by the considered patterns and the margin is computed on a separated testing set.
Implementation: When a limited number of examples is present in the dataset,

```
Input: A dataset T with labels in A ∈ {a_1, .., a_m}
       A classification function f
Output: An Activity Proximity Matrix M ∈ R^{m×m}
Split T in training set T_train and testing set T_test
foreach i=1,..,m do
    foreach j=1,..,m do
        Train f on T^{i,j}_train, with T^{i,j}_train ⊆ T_train s.t. T^{i,j}_train has only training points
        in {a_i, a_j}
        Compute the classification margin M(i, j) for class a_i and class a_j on T^{i,j}_test
    end
end
```

Algorithm 1: Compute Activity Proximity

the resulting testing set may be not representative of the activity distribution. The Random Forest classification function [14] is based on a supervised ensemble methodology that aggregates decisions trees trained on randomly chosen bootstrap samples of the training set. Hence, the set of examples not used for training can be used to obtain an estimate of the classification margin without the need of using an explicit testing set. For binary classification problems, the margin provided by Random Forest is equivalent to the geometrical classification margin as previously described [15].

4.2 Grouping Activities into Events

Method: Given the proximity measure provided by the margin, the aggregation of atomic activities is generated using the clustering capabilities of the Minimum Spanning Tree (MST) [16]. The spanning tree provides an acyclic graph containing as vertices the atomic activities in the set A. The graph is built in order to minimize the sum of the weighted edges, i.e., the classification margins, along its path. In the graph, the edges with lower weights connect the atomic activities that are contiguous and exhibit small margin. Cutting the edges relative to the highest margins provides a partition of the graph in groups. The resulting connected components represent the events generated. Algorithm 2 describes the steps used in the grouping procedure.
Implementation: In the algorithm, the optimal value of the threshold θ_w can be easily found considering the discrete distribution of all the values of the margin.

Input: A proximity matrix $M \in \mathbf{R}^{m \times m}$
A threshold θ_w
Output: A set of events E
Compute MST(N,E) Minimum Spanning Tree on M
Cut the edges with weight $w > \theta_w$
Set E as the connected components in the resulting graph

Algorithm 2: Clustering Activity using Minimum Spanning Tree

The threshold value is given by the margin value that provides the highest gap in the distribution.

4.3 Select Relevant Events

Method: Depending on their local position and sensing modalities, different sensors cannot sense all the atomic activities in the set A in the same way. In particular, patterns related to activities or events not directly sensed by the local node can be interpreted as noise from the detector: they will not contribute in the final recognition of the composite activities in C. Therefore, a ranking procedure measuring the importance of the discovered events is useful to establish which are the events that are significant for each detector. Stacked generalization [17] is a pattern recognition scheme that maximizes the classification performance of one or more classifiers using a multi-level architecture. The architecture feeds information from a set of base classifiers, called *level-0* classifiers, to a subsequent *level-1* classifier that provides the final decision. The space where the level-1 classifier works is called the reduced space as it is constituted by the predictions of the level-0 classifiers. These predictions will be the training examples for the level-1 classifier. Using this classification scheme, events relevance can be measured as a features selection process at the level-1 classifier.

Algorithm 3 describes the steps needed for this task using a general classification function f. Once the level-0 classifier is fit on the training set with labels E, the level-1 classifier is trained on the events prediction performed by the level-0 classifier on the test set that represent the features in this step. During the training of the level-1 classifier, the ranking is computed. Once events have been ranked based on their discriminant power for the final classification of the composite activity, significant events are chosen as the ones that provide at least 90% of the total importance in the ranking.

Implementation: As in the previous case, when a limited number of examples is considered, the Random Forest classification function can be used for computing feature importance on the samples not used in training. The Random Forest features importance measure is computed considering the mean increment in the classification error when a randomly selected features is changed in the tree. If the random permutation of the considered feature over all the trees provides an increment of the classification error, the feature represents an important variable in the classification process. The mean value of the incremental error, averaged over all the trees, provides the final measure of the importance.

Input: A dataset T with labels in $D \in \{E_1, .., E_t\}$ and $C \in \{c_1, .., c_n\}$
A classification function f
Output: A set of significant events E_s
Split T in training set T_{train} and testing set T_{test}
Train f^0 on T_{train} with labels E
Evaluate f^0 on T_{test} providing predictions E^{pred}
Train f^1 on E^{pred} with labels C and rank the best features in E^{pred}
Return E_s as the ordered set of events that provides the 90% of the total ranking importance

Algorithm 3: Compute Events Importance

This measure is in agreement with importance measures computed using linear regressors [18]. Nevertheless, Random Forest importance measure can be also computed for ill-posed problems where the number of data-points is much lower that the dimensionality of the features space.

5 Validation Methodology

The proposed methodology has been validated on a dataset collected in a car assembly scenario. The dataset used and the data collection process are briefly described in Sec. 5.1. The validation protocol and the performance measures adopted are described in Sec. 5.2

5.1 Dataset Description

A dataset collected using a car body installed in a laboratory environment has been used to gather data related to car assembling tasks. A total of 42 atomic activities have been collected using 12 distributed sensors worn by the workers and attached to different tools and parts of the car. Nine wireless sensor nodes have been used to record motion of different car parts. Two cordless automatic screwdrivers and a socket wrench have been used as tools for the assembly. Three wired sensors have been attached to a jacket at the wrist position of both right and left lower arms and the upper back. Two workers performed 10 repetitions of all the tasks. Experts manually annotated a total of 49 different detector events derived from the atomic activities for all sensor-detector nodes. Simple time-domain features are computed from raw 3-D accelerometer sensor data. The features includes sums and absolute sums, first and second deviations, minimum, average, and maximum amplitudes. The complete list of sensor nodes with their acronyms alongside the list of composite activities present in the dataset are shown in Table 1. A detailed description of the dataset can be found in [19].

5.2 Validation Protocol and Performance measure

Recall has been used to quantify the performance of the process and to evaluate the number of relevant atomic activities that the proposed configuration methodology is able to retrieve. The manual annotations and detectors configurations provided by experts have been considered as *ground-truth*. Recall is then

Table 1: Sensor nodes and composite activities in the car assembly scenario

Sensor Nodes	Right Lower Arm (RLA),Left Lower Arm (LLA),Central Upper Back (CUB),Front Light (FLIGHT),Brake Light (BLIGHT),Driver 1,Driver 2,Front Door (FDOOR),Back Door (BDOOR),Rattle
Composite Activities	Hood Rod,Mount Back Door,Mount Bar,Mount Brake Light,Mount Front Door,Mount Light,Mount Water tank,Test Back Door,Test Front Door,Test Hood,Test Trunk

expressed as the ratio between the number of annotated relevant activities that have been retrieved and the total number of annotated activities. A percentage of 100% of recall means that all the annotated activities relevant for the specific detector have been successfully retrieved.

In order to evaluate the coherence between manual and inferred configurations, the *events* distance has been used as performance measure. The events distance is defined as the minimum number of insertions, deletions and substitutions required to convert one event present in the generated configuration into the correspondent manually configured event. A zero value of events distance means that the events considered are constituted by the same atomic activities.

A 3-folds cross-validation approach has been used for evaluating the methodology. At each step of the cross-validation approach, one fold has been used for inferring the detectors structure, one fold has been used for training the classification architecture and one fold for testing purpose. A total of 3 runs of cross-validation has been performed, for a total of 9 experiments. Final performance measures are obtained by averaging all results. Classification accuracy has been adopted as recognition performance measure. In all the experiments, the Random Forest classifier has been trained on 151 classification and regression trees.

The K-Means clustering algorithm has been used for comparison purposes. Being the number of events not previously know, the procedure described in [20] has been used for automatically evaluate the number of clusters.

6 Experimental Results

Experimental results and discussions on aspects of interest are reported in the following subsections. In particular, results of the configuration process are reported in 6.1 for both k-Means and the proposed methodology, denoted as *MST*. Recognition performances for both detectors and overall classification architecture are reported in 6.2. Finally, 6.3 presents quantitative results in terms of reduction of computational resources that the inferred configuration provides for each detector.

6.1 Detectors Configuration and Comparison with Experts Annotations

Activity recall is reported in Fig. 3 for all the sensor nodes. For the wide majority of nodes, the MST methodology is able to retrieve all the relevant activities per detector as annotated by experts. RLA and LLA exhibit the lowest recall. In particular, LLA exhibits a recall of less than 50%. On the other hand, k-Means is able to retrieve all the annotated activities only for the Trunk, Blight and Rattle sensor nodes. The mean value of the events distance computed over all the events

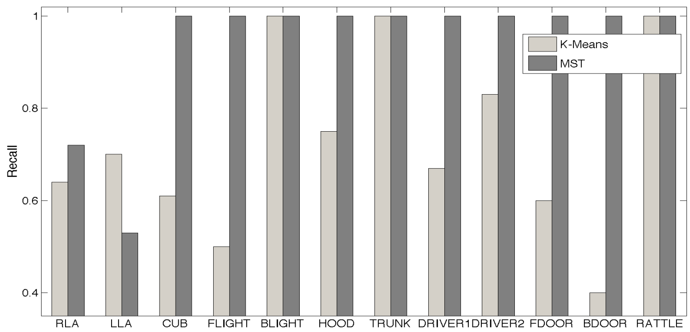

Fig. 3: Recall obtained for k-Means and MST: the value measures the percentage of relevant activities retrieved with respect to manual annotated for each detector node

inferred in the detectors configuration is shown in Fig. 4. The MST methodology infers configurations that are very similar to the configurations manually set by experts. In most of the cases, the events discovered coincide with the manually annotated events. Nevertheless, the configuration generated with the k-Means grouping shows very high values of distance indicating events that significantly deviate from the annotations. This is principally due to the fact that the algorithm collects data-points of contiguous activities in the same cluster generating events with elements derived from many different atomic activities. Both configurations generated by k-Means and MST contains events that experts did not annotate. This fact is exemplified in Fig. 5 where the configurations inferred for Trunk and Brake Light are reported. Both configurations identify one event that experts did not annotate. Nevertheless, the mutual presence of the events reflects the possibility that the activities of the nodes are related. Nevertheless, these hidden patterns cannot completely discriminable for the detector. This case and other similar ones obtained for different nodes, motivate the low recall measure obtained in the RLA, LLA and CUB detectors: in those nodes, the amount of activity patterns present makes extremely difficult an accurate manual grouping. Moreover, this fact represents the motivation why no null events distance is obtained. Finally, it is worth to be noted that, being the presence of these

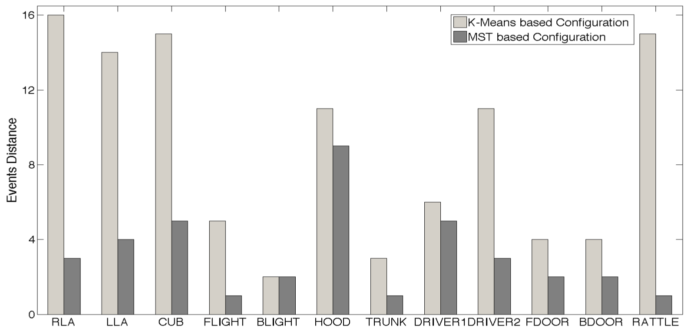

Fig. 4: Events distance obtained by k-Means and MST configurations: small values of events distance show that the configuration generated is consistent with the manually configured detectors

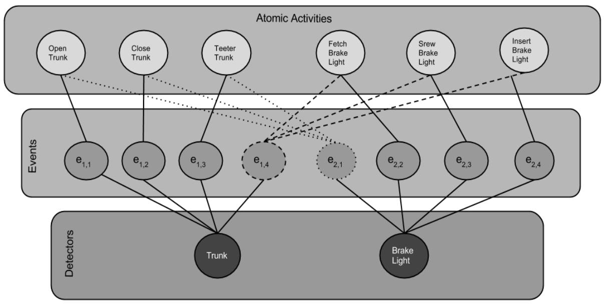

Fig. 5: Example of Detector Configuration for Trunk and Brake Light sensor nodes: dotted lines identify events not annotated by experts

non-annotated events a general case for all the detectors, measures like precision would exhibit very low performance value.

6.2 Classification Results

Events detection accuracy is reported in Fig. 6. Results derived by the manual and MST-based configuration indicate detection performances that are always higher than 99%. Although lower, detection accuracy derived by k-Means-based configuration still maintains a satisfactory level of performance. Detector performance without grouping are generally lower than 75% of accuracy and have been not reported. These results shows the capability of the AED paradigm to significantly boost the detection performance in the sensor node. This enhancement is also reflected in the final classification step where composite activities are considered. Mean value and standard deviation computed over all the composite activities are reported in Fig. 7. Starting from a classification accuracy of 73% obtained without configuration, accuracy reaches 78% and 84% when configuration based on the manual and k-Means grouping methodologies are used respectively. The highest performance is achieved using detectors configured with the MST configuration with a classification accuracy of 88%, corresponding to a performance boost of 15% with respect the baseline performance. This significant improvement can be easily understand when looking in depth into the detector configuration process. The AED paradigm aims to find a maximum margin classifier that easily separates the activity patterns in the features space of the detector, disregarding the actual classes provided by the atomic activities. In particular, the MST configuration explicitly groups data looking for the maximal margin distance. Due to this behavior, the classification is significantly boosted towards high level of recognition performance. Based on the same principle, the complexity of the detector will be also significantly reduced thanks to the possibility of learn simple separation boundaries, as shown in the following subsection.

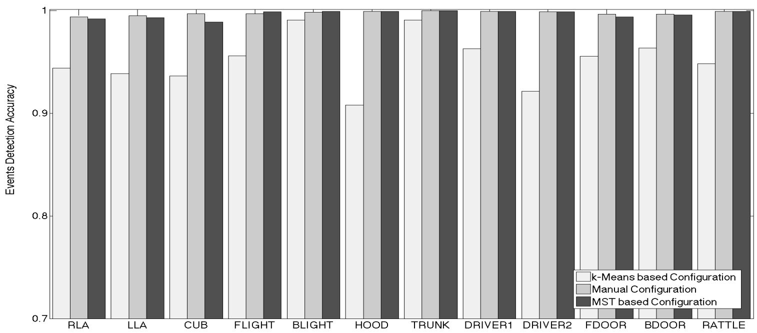

Fig. 6: Events detection accuracy for k-Means,manual and MST configurations: for the manual and MST grouping, detection accuracy reaches 99% for all the the sensors. Baseline accuracy obtained without configuration is 75%

6.3 Reduction in Detectors Computational Resources

Detectors have been modeled by means of classification and regression trees (CART) composing the Random Forest classifier. Hence, a simple measure of complexity for detectors is provided by the number of nodes each CART is composed. This complexity measure gives an idea of the complexity of the dataset that the tree is modeling. Events that are simple to model are described by very simple CARTs: in these trees the splits, implemented by if-then rules, represent the detection boundaries that the learning algorithm use to model the events. The mean number of nodes computed over all the CARTs classifier is reported in Fig. 8, for each detector and all the grouping methodologies. For comparison purposes, the mean number of nodes when no configuration is applied, is also reported. For all the inferred configurations, the mean number of nodes is significantly lowered. In particular, for the CUB detector, while approximately 470

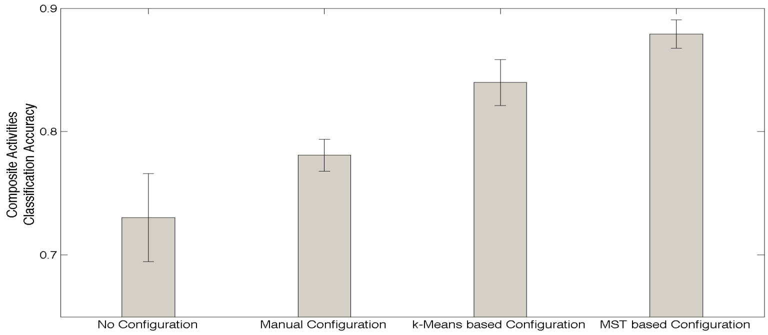

Fig. 7: Composite activities classification accuracy: mean value and standard deviation are reported for the the three grouping methodologies in comparison with the classification accuracy of the recognition process without grouping. The MST grouping provides a performance boost of 15%

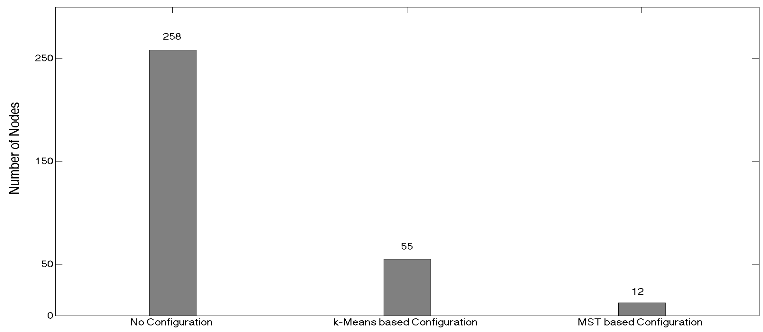

Fig. 8: Mean number of nodes in CARTs for each detectors: the number of nodes of the configured detectors is significantly lower than in the baseline classifier

nodes are needed for modeling the detector without events, a mean number of 100 and 29 nodes are needed for the k-Means and MST configurations. Using the MST grouping methodology, detectors modeling events in Fligth, Blight, Trunk and Rattle sensor nodes use CARTs with less than 10 nodes. As previously stated, this behavior is symptomatic of the construction of maximum margin classifiers: few decision boundaries are enough for obtaining a consistent and powerful discrimination between events.

7 Conclusions

In this paper, a data-driven methodology for inferring the configuration of distributed activity detectors has been presented and validated. The methodology, based on the framework of Activity-Events-Detector (AED), learns groups of activity patterns that look similar from the local perspective of the sensor nodes and are relevant for the final recognition of composite activities. The methodology provides an high-level grouping of the activities that does not depend directly by activity data-points and makes no assumptions regarding the underlying data distributions. In addition, relevant groups of activities are discovered using a ranking procedure based on a features selection strategy in a hierarchical classification architecture. Comparative results with manual annotations show that the detectors are configured coherently to the configuration provided by experts. In addition, the methodology generates groups of activities that, although not present in the manual configuration, significantly help in the recognition tasks. Experimental results obtained, validated on a multiple-runs cross-validation approach, show that the configuration provided by the proposed methodology significantly boosts the recognition performance at both detector and architectural level. In particular, all the events are detected with accuracy generally higher than 99% and the classification performance of composite activities is significantly enhanced. Furthermore, results obtained show that simple decision boundaries are needed when the AED paradigm is applied and the correspondent detectors are significantly simplified in terms of computational constraints.

Future works aim to further generalize the methodology and deepen understanding in all its aspects. Although a cross-validation scheme is used for providing statistical correctness to the results obtained, an exhaustive series of experiments over multiple datasets is needed in order to study the behavior of the methodology under different scenarios. In particular, a more complex set of activities should be considered in order to gain insights about the limitations of the methodology, specially when highly dense datasets are considered. Although not explicitly tested, the methodology has been developed in order to provide robustness to noise. Nevertheless, a comprehensive study related to testing the behavior of the methodology under different noise levels is needed in order to provide a further generalization step. Last but not least, a thorough study should be considered specially in relation to the theoretical aspects provided by the AED framework and the methodology proposed. Results obtained suggest that the theory of maximum margin classifiers can play a significant role in the further theoretical development of the AED paradigm.

Acknowledgments. This work was supported by the EU FP7 Marie Curie Network iCareNet under grant number 264738.

References

1. P. Rashidi, A. Mihailidis, A survey on ambient assisted living tools for older adults, IEEE Journal of Biomedical and Health Informatics 17 (3) (2013) 579–590.
2. A. Schmidt, Context-Aware Computing: Context-Awareness, Context-Aware User Interfaces, and Implicit Interaction, 2013.
URL `http://www.interaction-design.org/encyclopedia/context-aware_computing.html`

3. O. Amft, C. Lombriser, Modelling of distributed activity recognition in the home environment, in: EMBC 2011: Proceedings of the 33th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, IEEE, 2011, p. 1781–1784.
4. C. Lombriser, O. Amft, P. Zappi, L. Benini, G. Tröster, Benefits of Dynamically Reconfigurable Activity Recognition in Distributed Sensing Environments, Vol. 4 of Atlantis Ambient and Pervasive Intelligence, World Scientific Publishing Co., 2010, p. 261–286.
5. J. Aggarwal, M. Ryoo, Human activity analysis: A review, ACM Comput. Surv. 43 (3) (2011) 16:1–16:43.
6. Y. Aloimonos, Sensory grammars for sensor networks, Journal of Ambient Intelligence and Smart Environments 1 (1) (2009) 15–21.
7. P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, G. Troster, Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness, in: 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2007, pp. 281–286.
8. A. M. J. Sarkar, K. Hasan, Y.-K. Lee, S. Lee, S. Zabir, Distributed activity recognition using key sensors, in: Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, Vol. 03, 2009, pp. 2245–2250.
9. H. Storf, T. Kleinberger, M. Becker, M. Schmitt, F. Bomarius, S. Prueckner, An event-driven approach to activity recognition in ambient assisted living, in: Ambient Intelligence, Vol. 5859 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 123–132.
10. T. L. M. van Kasteren, G. Englebienne, B. J. A. Kröse, Hierarchical activity recognition using automatically clustered actions, in: Proceedings of the Second international conference on Ambient Intelligence, AmI'11, 2011, pp. 82–91.
11. A. K. Jain, Data clustering: 50 years beyond k-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.
12. C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
13. B. Scholkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2001.
14. L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
15. L. Breiman, Some infinity theory for predictor ensembles, Tech. rep. (2001).
16. C. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, Computers, IEEE Transactions on C-20 (1) (1971) 68–86.
17. D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.
18. G. Ulrike, Variable importance assessment in regression: Linear regression versus random forest, The American Statistician 63 (4) (2009) 308–319.
19. O. Amft, C. Lombriser, T. Stiefmeier, G. Tröster, Recognition of user activity sequences using distributed event detection, in: European Conference on Smart Sensing and Context (EuroSSC), 2007, pp. 126–141.
20. U. von Luxburg, Clustering stability: An overview, Foundations and Trends in Machine Learning 2 (3) (2009) 235–274.